



# Red Hat OpenShift GitOps 1.12

## Argo CD 应用程序

使用 Argo CD 仪表板、oc 工具或 GitOps CLI 在 OpenShift 集群上创建和部署应用程序。



## Red Hat OpenShift GitOps 1.12 Argo CD 应用程序

---

使用 Argo CD 仪表板、oc 工具或 GitOps CLI 在 OpenShift 集群上创建和部署应用程序。

## 法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 摘要

本文档提供有关使用 Argo CD 仪表板、oc 工具或 GitOps CLI 创建和部署应用程序的说明。它还讨论如何在 Argo CD 中验证自我修复行为。

---

## 目录

<b>第 1 章 使用 ARGO CD 部署 SPRING BOOT 应用程序</b> .....	<b>3</b>
1.1. 使用 ARGO CD 仪表板创建应用程序	3
1.2. 使用 OC 工具创建应用程序	3
1.3. 验证 ARGO CD 自助行为	4
<b>第 2 章 使用 GITOPS CLI 创建应用程序</b> .....	<b>6</b>
2.1. 使用 GITOPS CLI 在默认模式中创建应用程序	6
2.2. 使用 GITOPS CLI 以核心模式创建应用程序	7
2.3. 其他资源	8



# 第 1 章 使用 ARGO CD 部署 SPRING BOOT 应用程序

使用 Argo CD，您可以使用 Argo CD 仪表板或使用 **oc** 工具将应用程序部署到 OpenShift Container Platform 集群。

## 1.1. 使用 ARGO CD 仪表板创建应用程序

Argo CD 提供了一个仪表板，供您创建应用程序。

### 先决条件

- 以管理员身份登录到 OpenShift Container Platform 集群。
- 您已在 OpenShift Container Platform 集群中安装了 Red Hat OpenShift GitOps Operator。
- 已登录到 Argo CD 实例。

### 流程

1. 在 Argo CD 控制面板中，单击 **NEW APP** 以添加新 Argo CD 应用。
2. 对于此工作流，使用以下配置创建一个 **spring-petclinic** 应用程序：

应用程序名称

**spring-petclinic**

project

**default**

同步策略

自动

仓库 URL

<https://github.com/redhat-developer/openshift-gitops-getting-started>

修订

**HEAD**

路径

**app**

目的地

<https://kubernetes.default.svc>

命名空间

**spring-petclinic**

3. 单击 **CREATE** 以创建应用程序。
4. 打开 Web 控制台的 **Administrator** 视角，再展开 **Administration** → **Namespaces**。
5. 搜索并选择命名空间，然后在 **Label** 字段中输入 **argocd.argoproj.io/managed-by=openshift-gitops**，以便 **openshift-gitops** 命名空间中的 Argo CD 实例可以管理您的命名空间。

## 1.2. 使用 oc 工具创建应用程序

您可以使用 **oc** 工具在终端中创建 Argo CD 应用程序。

### 先决条件

- 您已在 OpenShift Container Platform 集群中安装了 Red Hat OpenShift GitOps Operator。
- 已登陆到 Argo CD 实例。

### 流程

1. 下载 [示例应用程序](#)：

```
$ git clone git@github.com:redhat-developer/openshift-gitops-getting-started.git
```

2. 创建应用程序：

```
$ oc create -f openshift-gitops-getting-started/argo/app.yaml
```

3. 运行 **oc get** 命令以查看所创建的应用程序：

```
$ oc get application -n openshift-gitops
```

4. 在部署应用程序的命名空间中添加标签，以便 **openshift-gitops** 命名空间中的 Argo CD 实例可以管理它：

```
$ oc label namespace spring-petclinic argocd.argoproj.io/managed-by=openshift-gitops
```

## 1.3. 验证 ARGO CD 自助行为

Argo CD 持续监控已部署应用程序的状态，检测 Git 中指定清单和集群中的实时更改之间的差别，然后自动更正它们。这个行为被称为自我管理。

您可以在 Argo CD 中测试并观察自我管理的行为。

### 先决条件

- 您已在 OpenShift Container Platform 集群中安装了 Red Hat OpenShift GitOps Operator。
- 已登陆到 Argo CD 实例。
- 已部署并配置 **app-spring-petclinic** 应用程序示例。

### 流程

1. 在 Argo CD 仪表板中，验证您的应用程序是否具有 **Synced** 状态。
2. 点 Argo CD 仪表板中的 **app-spring-petclinic** 标题，查看部署到集群中的应用程序资源。
3. 在 OpenShift Container Platform web 控制台中进入 **Developer** 视角。
4. 修改 Spring PetClinic 部署，并将更改提交到 Git 仓库的 **app/** 目录。Argo CD 将自动将更改部署到集群。
  - a. Fork [OpenShift GitOps getting started repository](#)。

- b. 在 **deployment.yaml** 文件中，将 **failureThreshold** 值改为 **5**。
- c. 在部署集群中，运行以下命令验证 **failureThreshold** 字段更改的值：

```
$ oc edit deployment spring-petclinic -n spring-petclinic
```

5. 通过修改集群上的部署并扩展到两个容器集来测试自我修复行为，同时在 OpenShift Container Platform Web 控制台中观察应用程序。
  - a. 运行以下命令修改部署：

```
$ oc scale deployment spring-petclinic --replicas 2 -n spring-petclinic
```
  - b. 在 OpenShift Container Platform Web 控制台中，请注意部署最多扩展两个 pod，并立即缩减到一个 pod。Argo CD 检测到与 Git 存储库的区别，并在 OpenShift Container Platform 集群中自动修复应用程序。
6. 在 Argo CD 仪表板中，点 **app-spring-petclinic** 标题 → **APP DETAILS** → **EVENTS**。EVENTS 选项卡显示以下事件：Argo CD 检测集群中缺少同步部署资源，然后重新同步 Git 存储库进行更正。

## 第 2 章 使用 GITOPS CLI 创建应用程序

使用 Argo CD，您可以使用 GitOps **argocd** CLI 在 OpenShift Container Platform 集群中创建应用程序。

### 2.1. 使用 GITOPS CLI 在默认模式中创建应用程序

您可以使用 GitOps **argocd** CLI 在默认模式中创建应用程序。

#### 先决条件

- 您已在 OpenShift Container Platform 集群中安装了 Red Hat OpenShift GitOps Operator。
- 已安装 OpenShift CLI(**oc**)。
- 已安装 Red Hat OpenShift GitOps **argocd** CLI。
- 已登录到 Argo CD 实例。

#### 流程

1. 获取 Argo CD 服务器的 **admin** 帐户密码：

```
$ ADMIN_PASSWD=$(oc get secret openshift-gitops-cluster -n openshift-gitops -o
jsonpath='{.data.admin\.password}' | base64 -d)
```

2. 获取 Argo CD 服务器 URL：

```
$ SERVER_URL=$(oc get routes openshift-gitops-server -n openshift-gitops -o
jsonpath='{.status.ingress[0].host}')
```

3. 使用 **admin** 帐户密码并用单引号括起来来登录到 Argo CD 服务器：



#### 重要

用单引号括起密码可确保 shell 不会误解特殊字符，如 **\$**。始终使用单引号括起密码的字面值。

```
$ argocd login --username admin --password ${ADMIN_PASSWD} ${SERVER_URL}
```

#### Example

```
$ argocd login --username admin --password '<password>' openshift-gitops.openshift-
gitops.apps-crc.testing
```

4. 通过列出所有应用程序，验证您是否可以在默认模式下运行 **argocd** 命令：

```
$ argocd app list
```

如果配置正确，则现有应用程序将使用以下标头列出：

#### 输出示例

■

```
NAME CLUSTER NAMESPACE PROJECT STATUS HEALTH SYNCPOLICY
CONDITIONS REPO PATH TARGET
```

- 以默认模式创建应用程序：

```
$ argocd app create app-spring-petclinic \
  --repo https://github.com/redhat-developer/openshift-gitops-getting-started.git \
  --path app \
  --revision main \
  --dest-server https://kubernetes.default.svc \
  --dest-namespace spring-petclinic \
  --directory-recurse \
  --sync-policy automated \
  --self-heal \
  --sync-option Prune=true \
  --sync-option CreateNamespace=true
```

- 标记 **spring-petclinic** 目标命名空间，使其由 **openshift-gitops** Argo CD 实例管理：

```
$ oc label ns spring-petclinic "argocd.argoproj.io/managed-by=openshift-gitops"
```

- 列出可用的应用程序，以确认应用程序已成功创建并重复命令，直到应用程序具有 **Healthy** 和 **Synced** 状态：

```
$ argocd app list
```

## 2.2. 使用 GITOPS CLI 以核心模式创建应用程序

您可以使用 GitOps **argocd** CLI 在 **core** 模式中创建应用程序。

### 先决条件

- 您已在 OpenShift Container Platform 集群中安装了 Red Hat OpenShift GitOps Operator。
- 已安装 OpenShift CLI(**oc**)。
- 已安装 Red Hat OpenShift GitOps **argocd** CLI。

### 流程

- 使用 **oc** CLI 工具登录到 OpenShift Container Platform 集群：

```
$ oc login -u <username> -p <password> <server_url>
```

### Example

```
$ oc login -u kubeadmin -p '<password>' https://api.crc.testing:6443
```

- 检查 **kubeconfig** 文件中是否正确设置了上下文：

```
$ oc config current-context
```

- 将当前上下文的默认命名空间设置为 **openshift-gitops** :

```
$ oc config set-context --current --namespace openshift-gitops
```

- 设置以下环境变量来覆盖 Argo CD 组件名称 :

```
$ export ARGOCD_REPO_SERVER_NAME=openshift-gitops-repo-server
```

- 通过列出所有应用程序, 验证您是否可以在 **core** 模式下运行 **argocd** 命令 :

```
$ argocd app list --core
```

如果配置正确, 则现有应用程序将使用以下标头列出 :

#### 输出示例

```
NAME CLUSTER NAMESPACE PROJECT STATUS HEALTH SYNCPOLICY  
CONDITIONS REPO PATH TARGET
```

- 以 **core** 模式创建应用程序 :

```
$ argocd app create app-spring-petclinic --core \  
  --repo https://github.com/redhat-developer/openshift-gitops-getting-started.git \  
  --path app \  
  --revision main \  
  --dest-server https://kubernetes.default.svc \  
  --dest-namespace spring-petclinic \  
  --directory-recurse \  
  --sync-policy automated \  
  --self-heal \  
  --sync-option Prune=true \  
  --sync-option CreateNamespace=true
```

- 标记 **spring-petclinic** 目标命名空间, 使其由 **openshift-gitops** Argo CD 实例管理 :

```
$ oc label ns spring-petclinic "argocd.argoproj.io/managed-by=openshift-gitops"
```

- 列出可用的应用程序, 以确认应用程序已成功创建并重复命令, 直到应用程序具有 **Healthy** 和 **Synced** 状态 :

```
$ argocd app list --core
```

## 2.3. 其他资源

- [安装 GitOps CLI](#)
- [基本 GitOps argocd 命令](#)